

## Reti di calcolatori

Prova scritta – 3 settembre 2012

### Parte 1

#### Esercizio 1.1 [12 punti]

Si vuole realizzare un servizio Web per la raccolta degli ordini di un ristorante take-away. Il servizio prevede le seguenti pagine:

1. Un form in cui l'utente può inserire i cibi/bevande che fanno parte dell'ordine da un menù a scelta multipla, l'ora del ritiro da un menù a scelta singola e il nome del cliente in un campo testuale. Le opzioni dei menù devono essere generate dinamicamente a partire da due array rispettivamente con i nomi dei cibi/bevande (si considerino "pizza", "patatine", "hamburger", "hotdog", "birra", "bibita") e degli orari (si considerino gli orari "12:00", "12:30", "13:00", "19:00", "19:30", "20:00"). Si assume che la quantità prevista nell'ordine per ogni cibo/bevanda selezionata sia pari a 1.
2. Una pagina di raccolta dei dati che memorizza sul server gli inserimenti fatti col form di cui al punto 1 nella stessa sessione di lavoro. Nel caso in cui si inseriscano più ordini per la stessa persona e lo stesso orario in più richieste distinte, le quantità degli elementi selezionati vengono aggiunte a quelle già presenti.
3. Una pagina di riepilogo che stampa una tabella con il numero di elementi ordinati per ogni opzione prodotto-ora del ritiro. Si riportano poi tutte le ore del ritiro per cui c'è il massimo numero di elementi ordinati.

Si scrivano il form e le due pagine di raccolta dati e riepilogo usando HTML/PHP. Si supponga di utilizzare il metodo POST nel form.

#### Esercizio 1.2 [4 punti]

Dato il seguente codice PHP

```
$a = '3.2';  
$b = 12;  
$v = array("a" => "1$a", "b" => '3$b');  
$out = $v['a']-$v['b'];
```

indicare quali sono le variabili definite nell'interprete dopo l'esecuzione del codice, con il loro tipo e valore.

#### Domanda 1 [7 punti]

Illustrare il modo con cui un server Web comunica al client l'esito di una richiesta HTTP nei vari casi che si possono presentare, mettendo anche in evidenza il ruolo dei campi di intestazione.

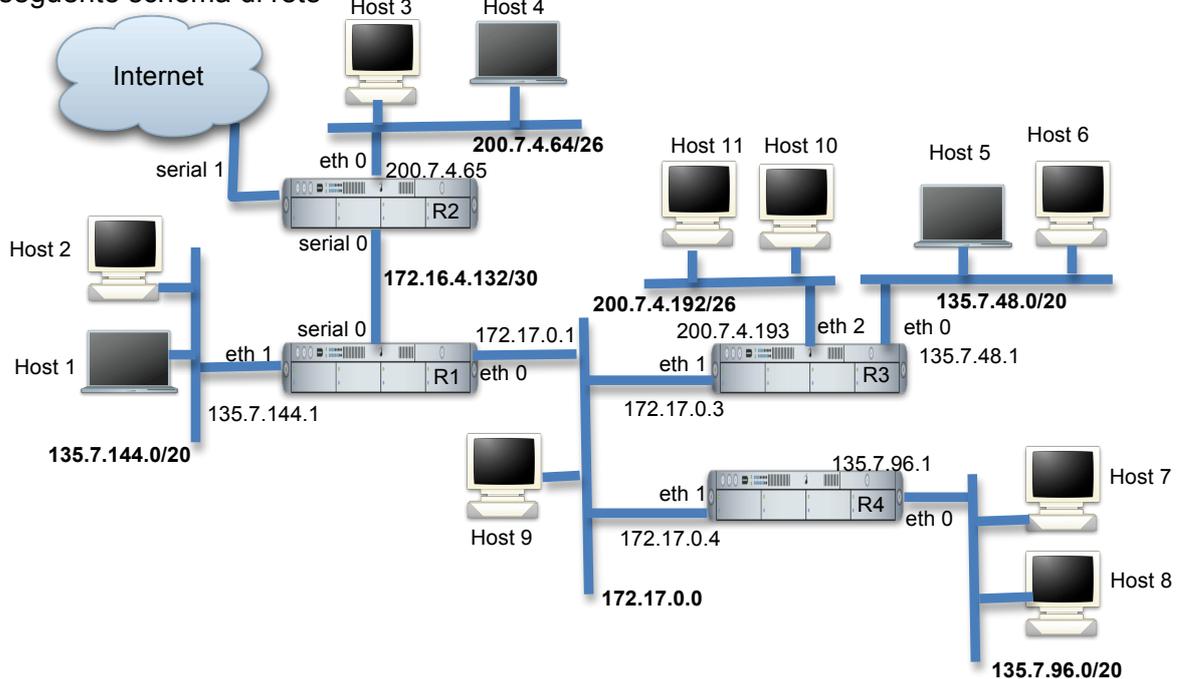
#### Domanda 2 [7 punti] (solo per chi deve recuperare la parte 1)

Illustrare i livelli principali della struttura gerarchica del sistema DNS specificando il ruolo dei server presenti nei vari livelli nel caso di un'interrogazione iterativa per la risoluzione di un nome.

## Parte 2

### Esercizio 2.1 [punti 6]

Dato il seguente schema di rete



scegliere la configurazione di rete dell'host 6 (IP, netmask e configurazione di routing) e indicare il contenuto delle tabelle di routing del router R1.

### Esercizio 2.2 [punti 8]

Si vuole definire un formato per l'interscambio dati relativo ai veicoli di una società di noleggio. Il file scambiato specifica la lista dei camper e la lista delle automobili. Per ciascun camper si indica la targa (string), la data di immatricolazione (date) e la lista degli optional presenti. Per ogni optional si indica il nome (string) e una descrizione (string). Per ciascuna automobile si specificano la targa (string), la classe (string) e il numero di posti (integer). Si proponga la struttura XML necessaria, mostrando un esempio, e si scriva il file XML schema associato.

### Esercizio 2.3 [punti 8]

Si scriva il codice javascript necessario ad implementare la seguente funzionalità in una pagina HTML: Inserendo un testo in un elemento in input di tipo text (elemento HTML `<input type="text">`), ogni volta che si preme il tasto invio (keyCode 13) il testo contenuto nel campo viene aggiunto come elemento (tag HTML `<li>`) di una lista predefinita (tag HTML `<ul>`) nel file HTML e il contenuto del campo di input viene svuotato (stringa vuota "").

### Esercizio 2.4 [punti 8]

Supponendo che un socket server TCP sia già stato opportunamente inizializzato e sia disponibile il suo identificatore nella variabile `sd` (`int sd`) scrivere il codice per implementare il seguente protocollo di comunicazione.

1. Dopo aver accettato una connessione, il server invia al client la stringa ottenuta dalla chiamata del metodo `char *getChallenge()`.
2. Il server attende una stringa in risposta e poi chiama la funzione `int checkAuth(char *key)` passando come parametro la stringa ricevuta.
3. Se la funzione `checkAuth()` produce un valore non nullo, chiama la funzione `char *getSecretMessage()` e invia la stringa ottenuta al client chiudendo poi la connessione. Se invece il valore di `checkAuth()` è 0 il server chiude subito la connessione.